

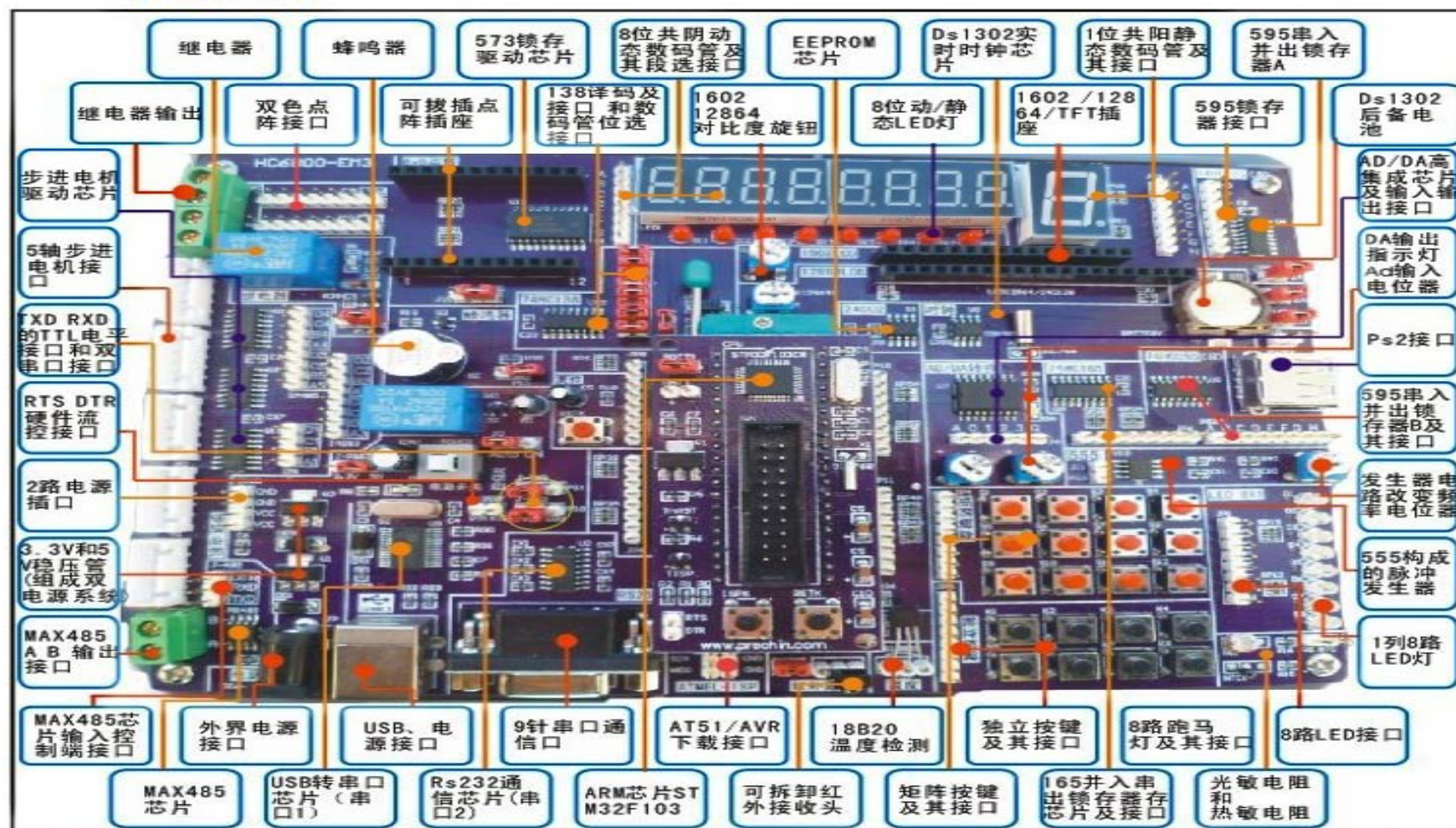
# ARM系统STM32嵌入式系统的开发实验

---

- ❖一 STM32简介及其实验环境
- ❖二 STM32F10X嵌入式系统实验
- ❖三 如何学习STM32
- ❖四 学习STM32嵌入式系统从GPIO口开始
- ❖五 注意事项

# 开发仪实验箱 部分硬件配置

## 功能简介



采用模块化分区，原理图清晰易懂，并带有注释，一看就懂 一学就会 一思就通

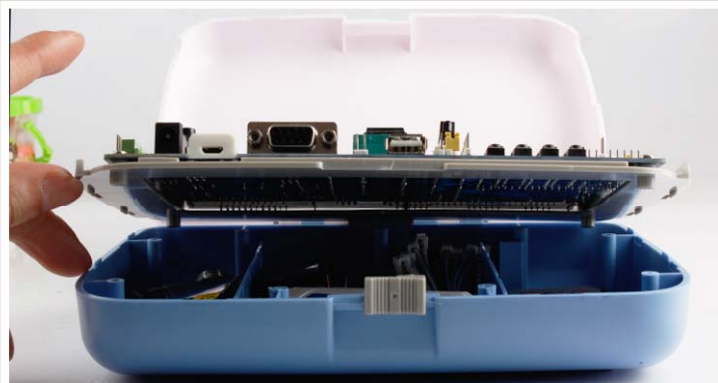


实物拍摄

Commodity reality



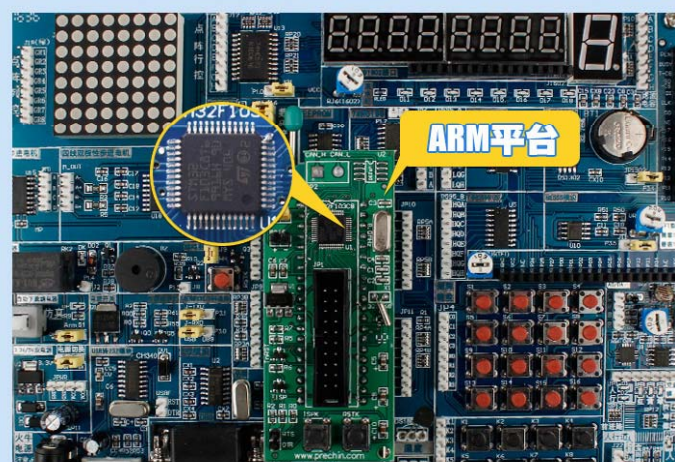
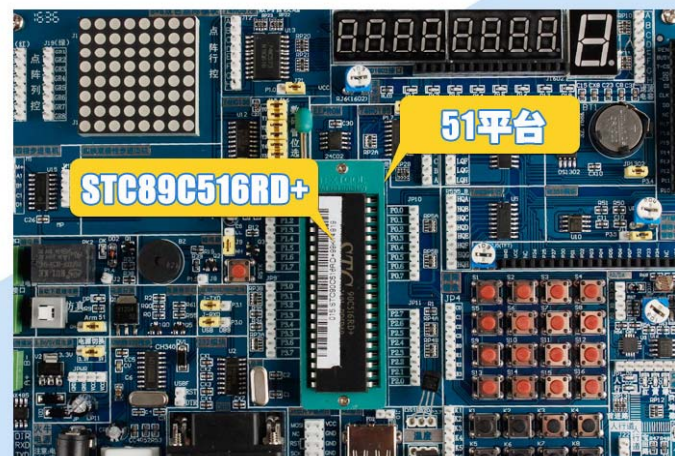
EM3-V2.2增强版



三种平台搭建

玩转开发板一板搞定

51平台 · ARM 平台 · AVR 平台

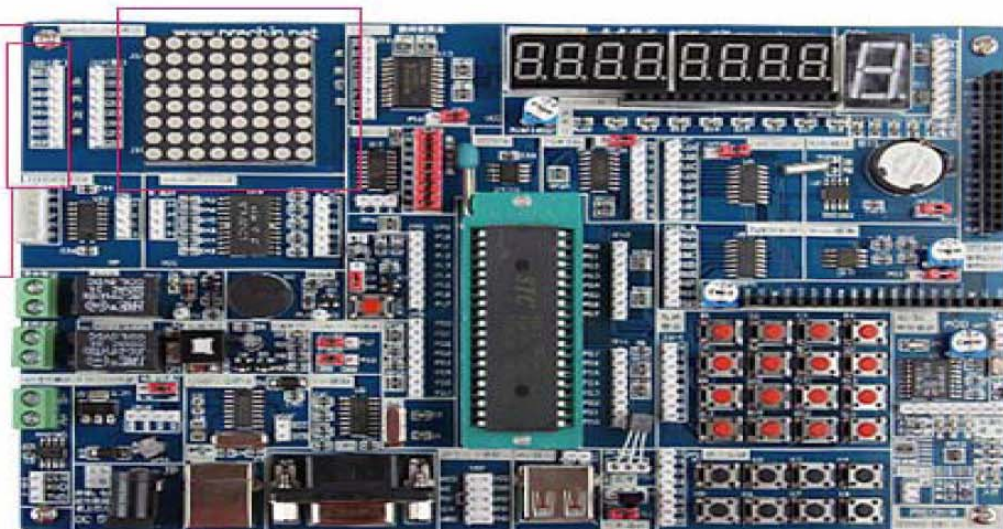


## 标准配置

1. 全新增强型STC89C516	1个
2. 优质USB数据线	1个
3. 八位排线	3条
4. 单P杜邦线	8条
5. 红色短路帽	19个
6. 光敏电阻,热敏电阻	各1个
7. DVD光盘	1张
8. 全新ABS材料塑料盒	1个
9. 2.0寸触摸彩屏	1个
10. 8X8LED点阵	1个
11. 步进电机驱动	1路
12. MINI12864	1个
13. 液晶屏mini1602	1个
14. 全新直流电机	1个
15. 全新红外接收头	1个
16. 全新红外遥控器	1个
17. 高速高矩进口步进电机	1个
18. 实验时钟电池座及电池	1PCS
19. 风扇	1个

## V2.2开发板 +

2.0



V3.0开发板正面图



# 一. STM32简介及其实验环境

- 硬件平台——开发仪
- 软件平台——Keil uVision集成开发环境  
集成开发环境提供了工程管理、程序编辑、代码下载、调试等所有功能
- 一般按以下流程来进行微处理器的实验：
  1. 建立工程；
  2. 选择目标器件；
  3. 配置工程属性；
  4. 打开/编写程序文件；
  5. 编译连接工程；
  6. 修改程序中的错误并重新编译连接；
  7. 下载程序至实验箱进行实验。
- 要使用Keil uVision2进行C8051F系列单片机和STM32F103系列单片机的在系统进行项目的编辑、汇编、编译、链接、下载和调试。



# 系统性能和运行环境

---

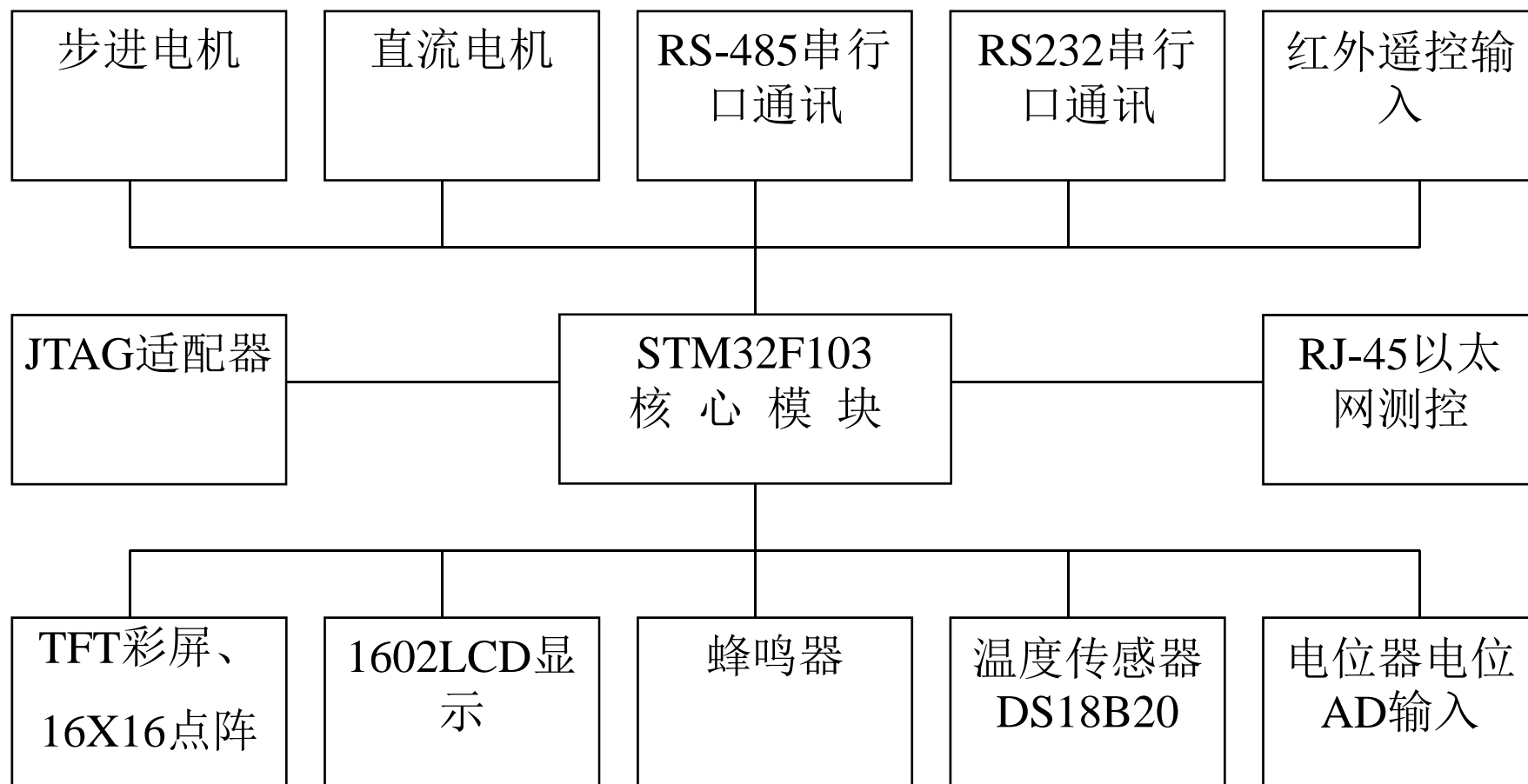
## ❖ 系统的主要性能指标：

- 系统时钟最高可达25MHz。
- 通过RS232接口（或USB）与PC机连接。
- 支持汇编语言、C51源代码和C语言级调试。
- 支持第三方软件工具（Keil C）。

## ❖ IDE 软件运行环境

- Windows 2000/XP操作系统以上版本
- 32Mb RAM
- 40Mb 自由硬盘空间
- 空闲的COM口，配USB接口的调试电缆可以用USB接口。

# 开发仪实验模块



## 开发仪实验箱模块资源介绍（1）

---

- ❖ 系统实验板除与CPU核心模块连接外，还配置了丰富的硬件资源，可根据需要安排多个实验内容。
- ❖ 系统实验板与核心模块组成实验的目标系统，然后由运行在PC机上的集成开发软件、JTAG口的协议转换器(EC2)，实现在系统开发调试。
- ❖ 实验者可参照本章后面的实验例程编译和下载各应用程序，也可独立设计程序(C或汇编)，然后就可开始在系统调试了。
- ❖ 该系统实验板由多个相对独立，又能组合的实验硬件模块组成。



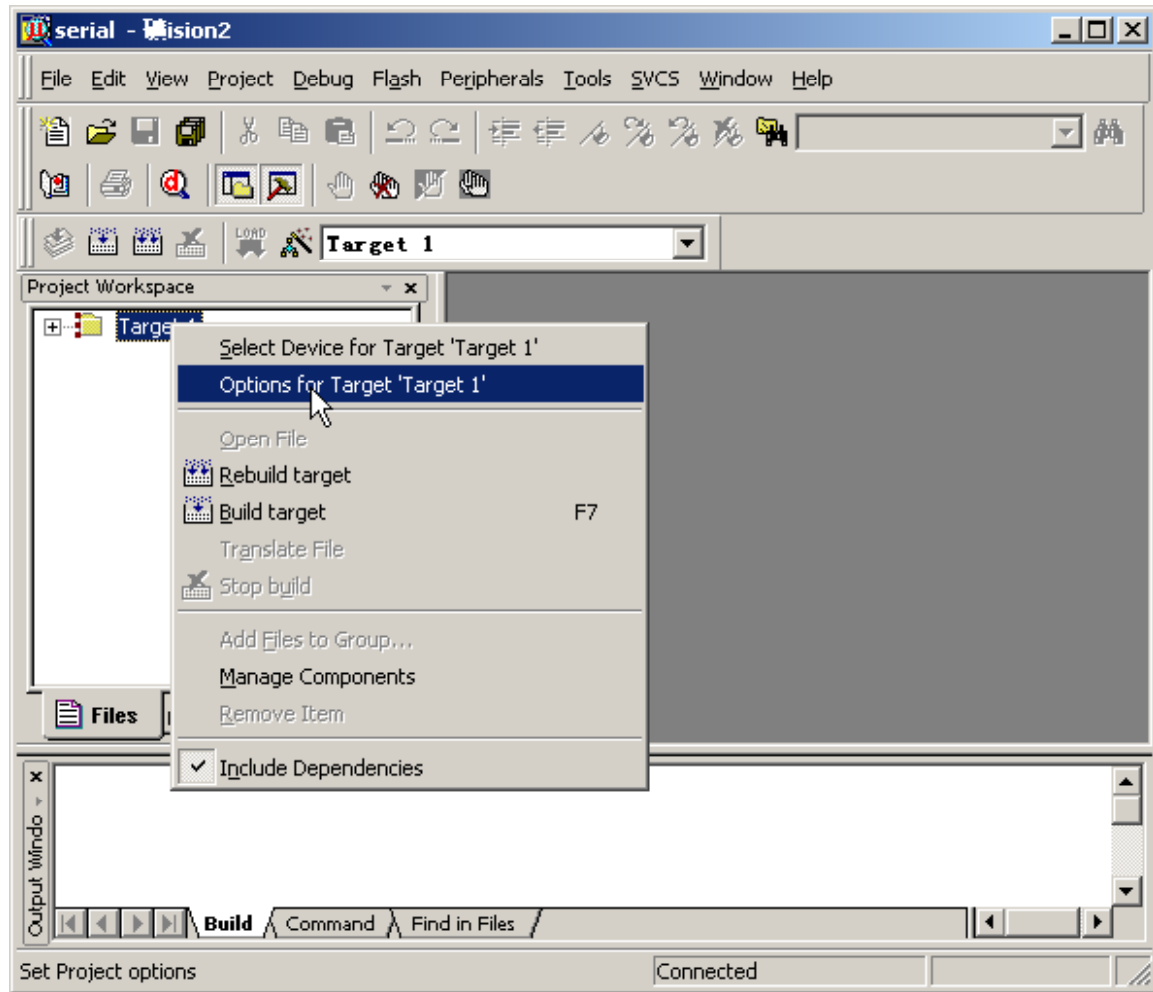
## 开发仪实验箱模块资源介绍（2）

---

- ❖(1) LCD1602模块，LCD数码显示；
- ❖(2) 开关量输出的D1~D16发光二极管；
- ❖(3) TFT彩屏；
- ❖(4) DS18B20数字温度传感器；
- ❖(5) 电位器测量电压模拟输入电路；
- ❖(6) 红外遥控器输入电路；
- ❖(7) 16×16点阵LED显示模块；
- ❖(8) RS-232通讯接口；
- ❖(9) RS-485通讯接口；
- ❖(10) RJ-45以太网接口；
- ❖(11) 直流电机及调速电路；
- ❖(12) 步进电机及控制电路；

# 目标处理器设置选项菜单

用Keil uVision2新建一个项目，然后在Keil uVision2集成开发环境左侧的Project Workspace的Files选项页中，选中“Target 1”，鼠标右键单击出现右键菜单，如右图所示，选择“Options for Target ‘Target 1’”菜单项，出现项目选项设置对话框。



## 二. STM32F10X嵌入式系统实验

---

### 课程教学目标

- ◆理解STM32F10X处理器及其嵌入式系统结构
- ◆了解STM32F10X处理器及其传感器和执行器操作
- ◆掌握STM32F10X的编程和调试方法



# 实验课实施计划

---

**实验总时数: 16学时，共4个实验内容**

**实验一** 利用STM32插件，进行C软件上传到开发仪观察按键检测和LED显示等传感检测和执行控制。

**实验二** ARM实验部分中第8个实验，利用STM32F103插件实现键盘扫描和LCD显示实验和调试。

**实验三** ARM实验部分中第19个实验，利用STM32F103插件实现DS18B20温度传感器的AD转换和LCD显示实验，检测、超温报警和记录温度变化。

**实验四** 参考51单片机高级实验中的智能计算器程序，利用STM32F103插件实现扫描键盘数字输入和独立键盘运算符输入以及LCD显示的计算器实验。

---

# 实验成绩考核办法

## ■ 实验课成绩计算：4个实验，实验课成绩满分100分

实验	1	2	3	4	总计
对应分数	25	25	25	25	100

## ■ 单个实验成绩考核方法：每个实验的成绩分为两个部分：一个是实验验收成绩，另一部分为实验报告成绩。

- 实验验收成绩：由任课教师根据学生的实验表现、实验结果的正确与否等因素给出评分，无教师确认成绩计为0。
- 实验报告成绩：根据报告的质量(完整性、正确性和工整性等)将验收成绩或向上、或向下、或保值不变作为本次实验的最终成绩，无报告，以报告部分计为0。

## ■ 实验课占课程期末总成绩30%

# 实验报告要求

---

1. 实验名称
2. 实验目的
3. 实验内容
4. 主程序流程图
5. 源代码（可以打印）
6. 程序调试过程
7. 执行结果



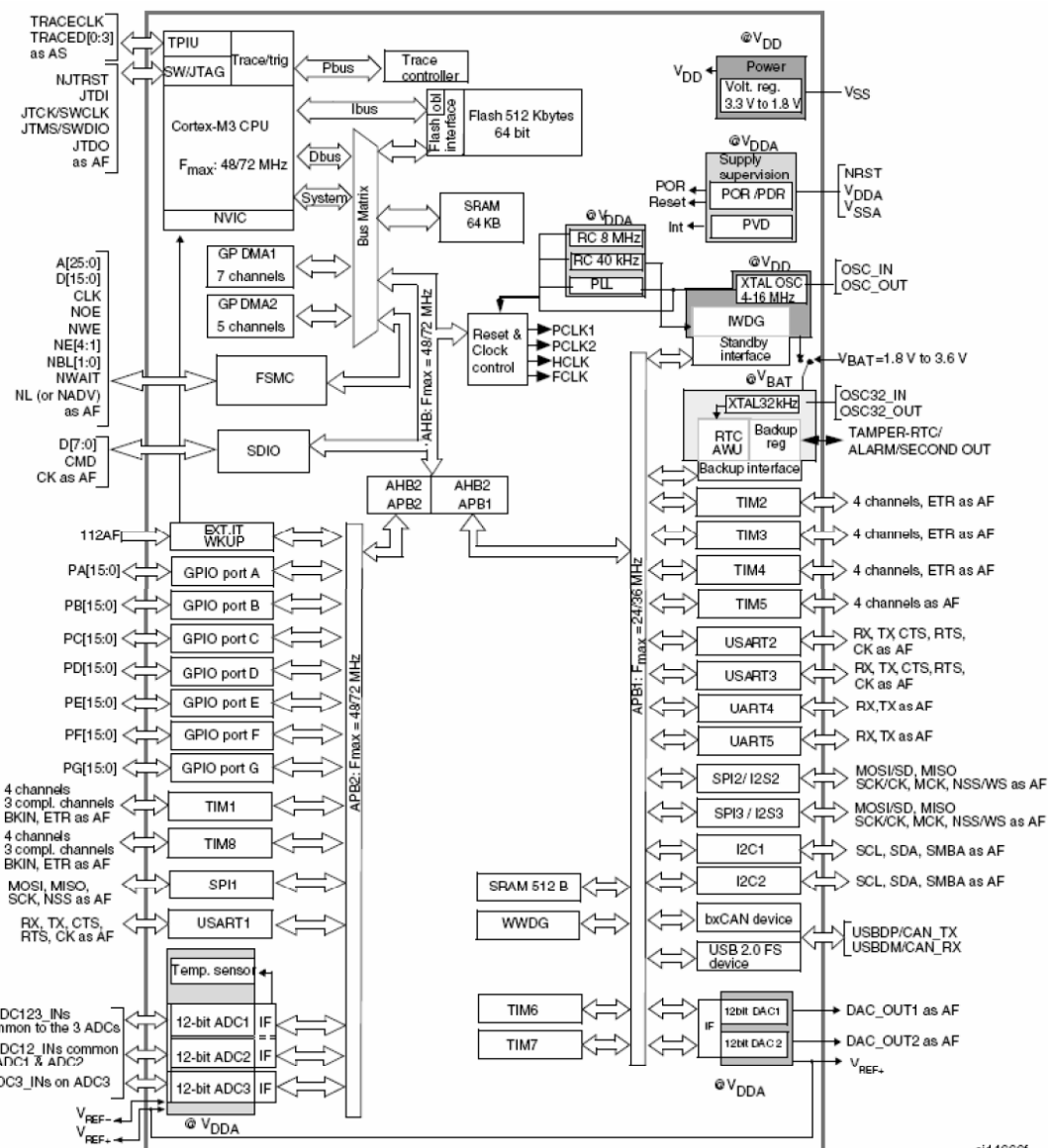
# 实验报告的要求

---

- ◆ 实验名称；姓名学号。
- ◆ 实验报告必须有下列内容：
  - 实验目的
  - 实验内容
  - 简要说明库函数作用，如
    - GPIO\_Init
    - GPIO\_SetBits
    - GPIO\_ResetBits
    - GPIO\_WriteBit
    - GPIO\_ReadInputDataBit
  - 实验任务
    - 实验原理
    - 主要编程要点
    - 主要程序代码
  - 实验结果及有关问题和解决方法
    - 问题1
    - 问题2

### 三. 如何学习STM32 ?

- 内部功能模块很多。  
**边学边试！ 边用边学！**
- 如果不用的模块，可不理会其存在。  
**用什么模块，学什么模块！**
- 对STM32有两种编程方法：
  - 寄存器编程  
STM32有庞大的寄存器组，直接对寄存器编程相当困难。
  - 函数编程  
**STM32有一大堆函数！**  
**边学边用！**  
**边用边学！**

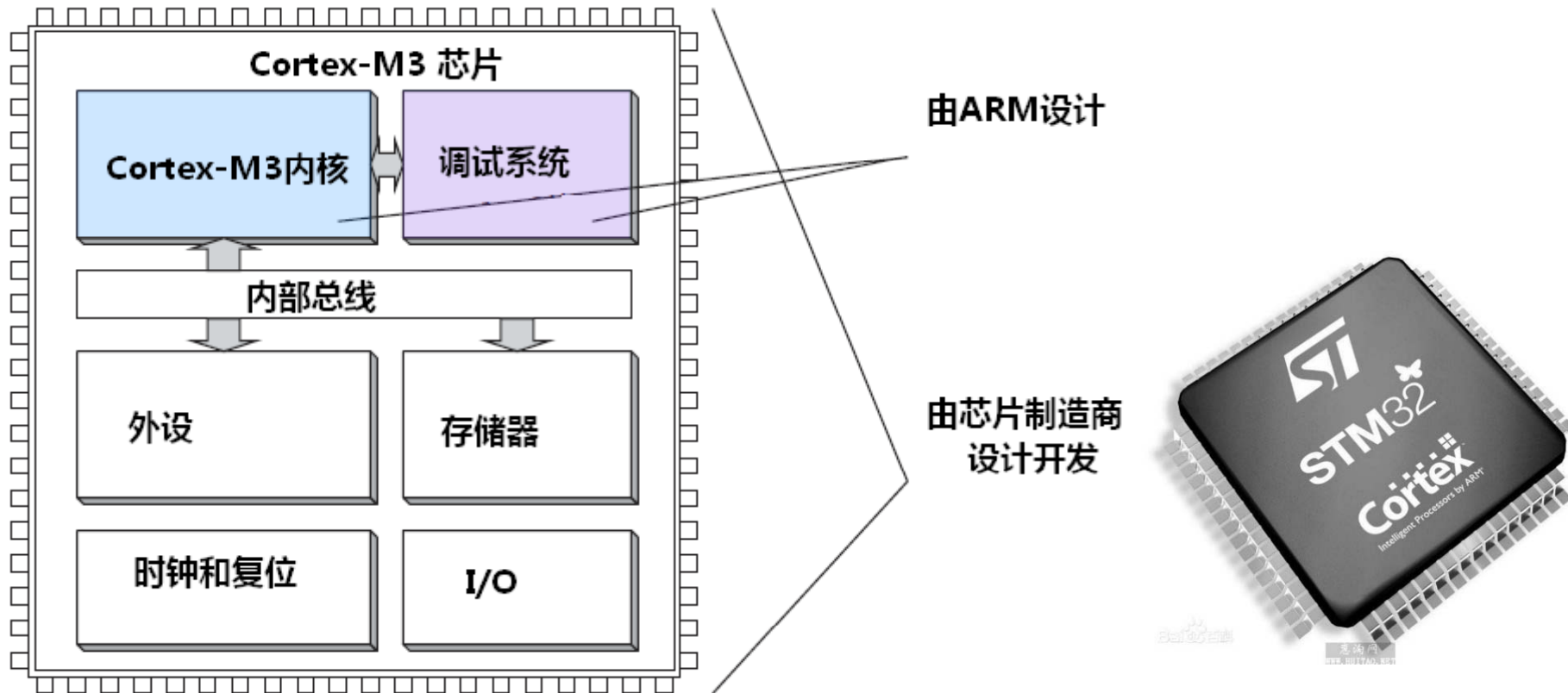


aj14666f

**学习过程中，养成反复查阅资料和教科书的习惯！**

# 主控芯片 STM32F103CBT6

- STM32F103CBT6是意法半导体公司生产的高性能、低功耗、低成本的嵌入式微控制器。
- 以ARM® Cortex™-M3为内核。





# STM32F103XXT6主要特点

---

- 以ARM® Cortex™-M3为内核。
- 具有大量的片上外设（多个通信口、多个定时器、全部I/O脚可设为中断模式、ADC、DAC、4个片内时钟、DMA等）
- 片内Flash程序存储器 128KB（CB），256KB（ZC）
- 片内SRAM 64KB
- 支持JTAG仿真和ISP下载
- 有免费的全套函数库，便于用开发



# STM32命名规则

开发仪主控芯片

STM32F103ZCT6

144脚

256KB程序存储器

STM32F103CBT6

48脚

128KB程序存储器



示例: STM32 F 103 C 8 T 6

产品系列

STM32 = 基于ARM®的32位微控制器

产品类型

F = 通用类型

产品子系列

101 = 基本型

102 = USB基本型, USB 2.0全速设备

103 = 增强型

引脚数目

T = 36脚

C = 48脚

R = 64脚

V = 100脚

Z = 144脚

闪存存储器容量

4 = 16K字节的闪存存储器

6 = 32K字节的闪存存储器

8 = 64K字节的闪存存储器

B = 128K字节的闪存存储器

C = 256K字节的闪存存储器

D = 384K字节的闪存存储器

E = 512K字节的闪存存储器

封装

H = BGA

T = LQFP

U = VFQFPN

Y = WLCSP64

温度范围

6 = 工业级温度范围, -40°C~85°C

7 = 工业级温度范围, -40°C~105°C

# STM32F10x : 两个现行的产品系列

两个系列都包含:

多达128K字节的闪存

3 x USART

2 x SPI

2 x I<sup>2</sup>C

3 x 16位定时器

4-16MHz 主振荡器

内嵌8MHz的RC振荡器和32kHz的RC振荡器

实时钟

2 x 看门狗

复位电路  
上电/断电复位  
电压检测

7通道DMA

80%管脚是通用I/O

## 增强型 **STM32F103**

72MHz  
CPU

多达  
20K字节  
SRAM

2个12位  
ADC(1μs)  
温度传感器

USB  
2.0  
全速

CAN  
2.0B

专用  
PWM  
定时器

+

## 基本型 **STM32F101**

36MHz  
CPU

多达  
16K字节  
SRAM

1个12位  
ADC (1μs)  
温度传感器

**STM32 32位微控制器**





## STM32F103CBT6的部分资源

7个16位通用IO口 — **PGIO**

- ▶ PA[0: 15]
- ▶ PB[0: 15]
- ▶ PC[0: 15]
- ▶ PD[0: 15]
- ▶ PE[0: 15]
- ▶ PF[0: 15]
- ▶ PG[0: 15]

➤ 所有GPIO脚可设为中断模式

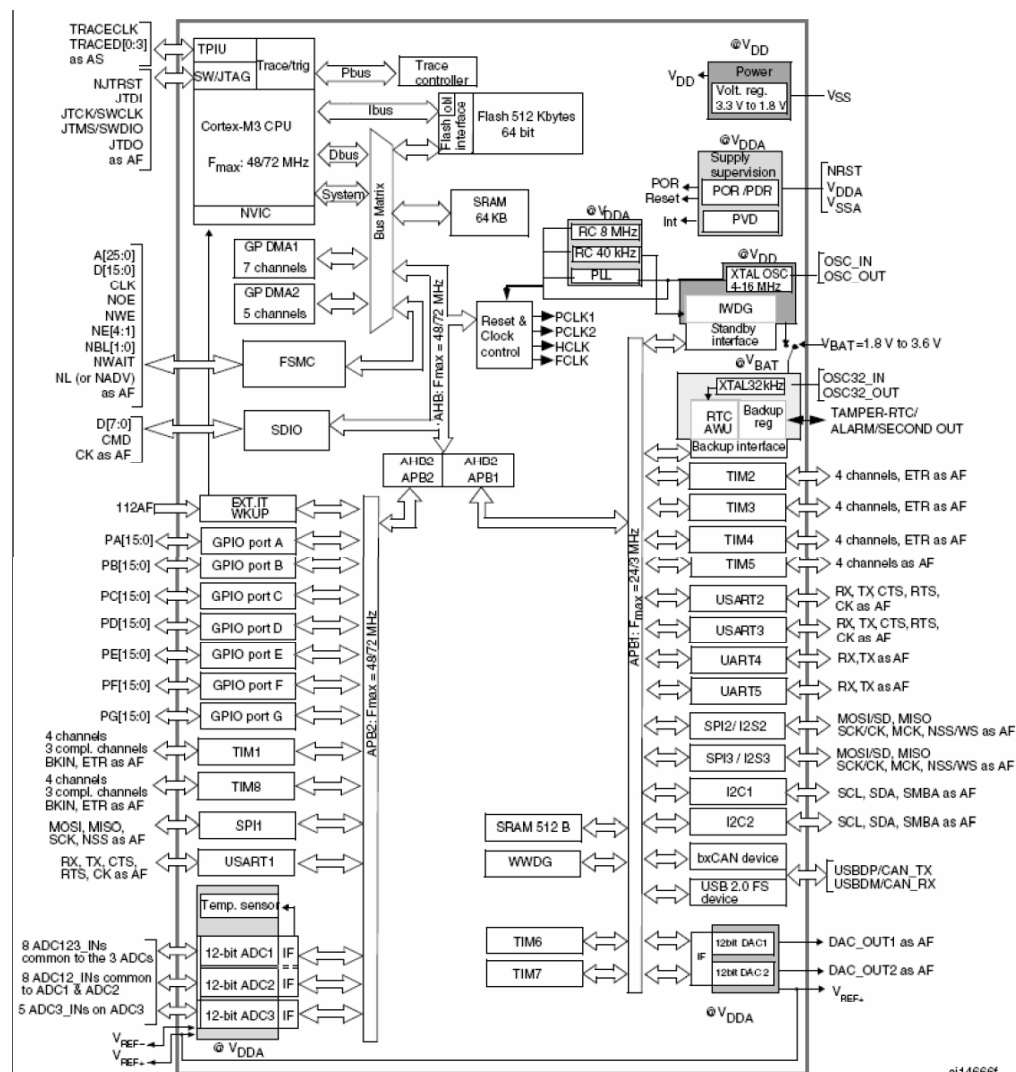
78个定时器

➤ 5个通信口

(SUB, USART, IIC, SPI, CAN)

ADC, DAC, DMA等

■ ■ ■ ■ ■ ■




# 参考资料：下载文件夹和电子手册

---

 STM32F10xxx\_库函数\_cn

**本手册介绍了STM32F103xx的固件函数库。**

反复查阅库函数，  
逐步掌握STM32的开发！

 STM32F10x内部寄存器描述\_cn

**用寄存器方式操作STM32，可参考此手册，适合于高级编程。**

 STM32F103CDE增强型系列数据手册(2009年3月)

**用于了解STM32增强型产品的管脚和片上资源。**

 STM32-IAR开发教程

**IAR建立工程的方法**

 STM32外设概述\_ppt\_cn

**STM32官方资料，初步了解STM32及开发方法**

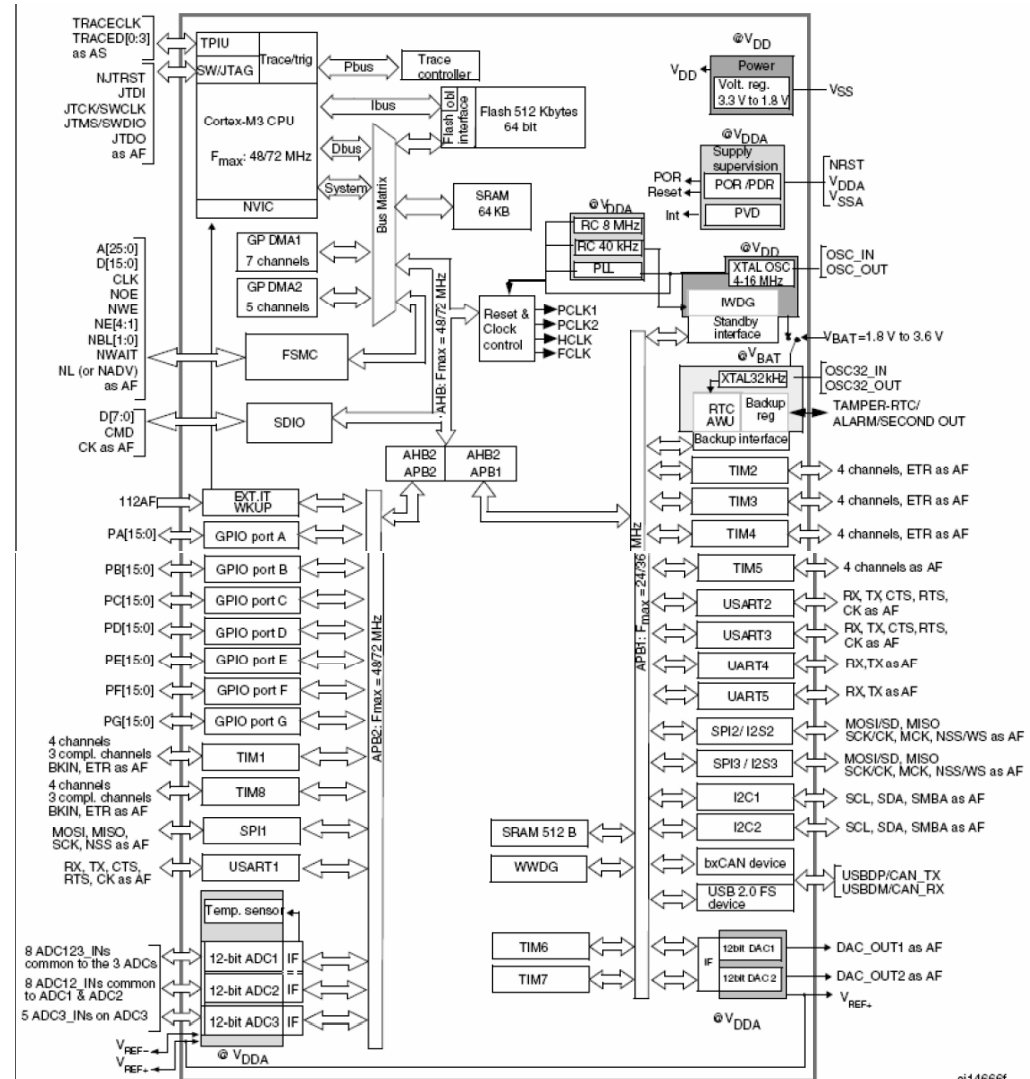
 STM32中文参考手册

**介绍STM32微控制器产品的功能模块结构**

**ARM系统开发仪的使用方法等手册。**

# 四. 学习STM32嵌入式系统从GPIO口开始

- 内部功能模块很多  
——边用边学！
- 如果不用的模块，  
可不理会其存在  
——用什么模块，  
学什么模块！
- 一般的学习顺序（使用频度）：  
I/O操作  
定时器应用  
RS232串行总线  
I2C串行总线  
AD、DA变换  
.....



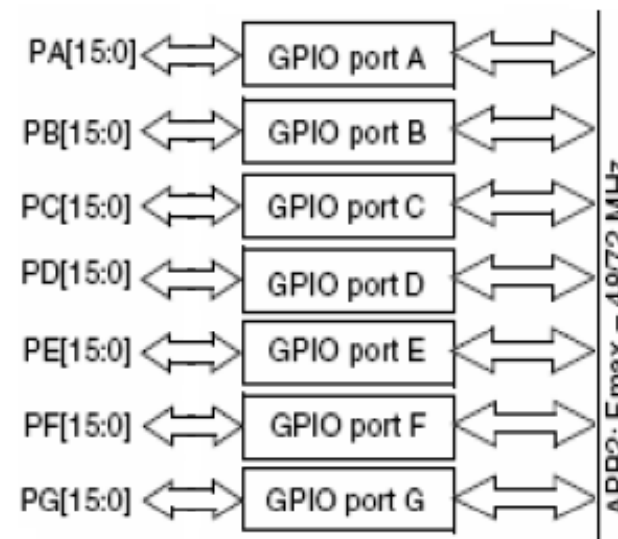
# STM32 IO端口及其操作

➤ STM32有2-7个通用I口GPIO

➤ 每个端口16位

- PA[15:0]
- PB[15:0]
- PC[15:0]
- PD[15:0]
- PE[15:0]
- PF[15:0]
- PG[15:0]

➤ 用户可利用库函数对GPIO进行操作



序号	函数名	描述
1	GPIO_Init	根据GPIO_InitStruct中指定的参数初始化GPIOx寄存器
2	GPIO_SetBits	设置指定的数据端口位
3	GPIO_ResetBits	清除指定的数据端口位
4	GPIO_WriteBit	设置或者清除指定的数据端口位
5	GPIO_ReadInputDataBit	读取指定端口管脚的数据
	.....	.....

# GPIO的8种IO模式

➤ 2-7个16位通用可编程口PGIO PA[15:0] ~ PG[15:0]

➤ GPIO的8种IO模式：

初始化时  
必须设置端口的IO模式

① 浮空输入\_IN\_FLOATING

② 上拉输入\_IPU ——IO内部上拉电阻输入

③ 下拉输入\_IPD ——IO内部下拉电阻输入

④ 模拟输入\_AIN—— ADC模拟输入

⑤ 开漏输出\_OUT\_OD——IO 输出0：接GND；输出1：悬空，需外接上拉电阻。

⑥ 推挽输出\_OUT\_PP——IO输出0：接GND；IO输出1：(接VCC)。

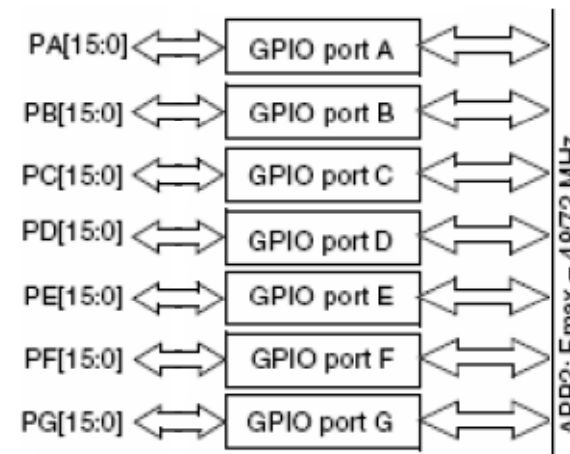
⑦ 复用功能的推挽输出\_AF\_PP——片内外设（I2C的SCL，SDA）

⑧ 复用功能的开漏输出\_AF\_OD——片内外设（TX1，MOSI，MISO，SCK，SS）

➤一般用“上拉输入”，“推挽输出”

# IO端口初始化函数 GPIO\_Init

- 使用IO口时必须先初始化IO口
- 与IO有关的函数和定义分布在：
  - STM32固件库文件 `stm32f10x_gpio.c`
  - 头文件 `stm32f10x_gpio.h`
- 如，IO口初始化函数：



```
GPIO_Init( GPIOx, &GPIO_InitStructure );
```

输入参数1: GPIOx  
x可以是A, B, C, D, E, F, G

函数原形见:  
《STM32F103xx 固件函数库》

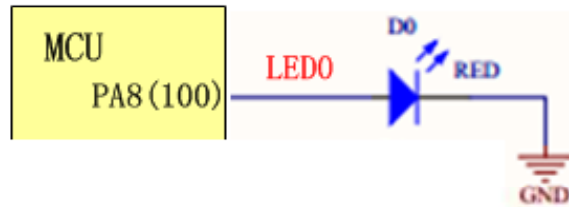
输入参数2,  
是初始化参数结构体指针GPIO\_InitStruct,  
其中包含了外设IO口的配置信息。



# 类的定义及初始化过程

- 在 “stm32f10x\_gpio.h” 头文件中（位于例程一：Library->inc->），先用typedef 声明了结构体类型

```
typedef struct
{ uint16_t      GPIO_Pin;    //IO引脚
  GPIOMode_TypeDef GPIO_Mode; //IO模式
  GPIOSpeed_TypeDef GPIO_Speed; //IO口速度
}GPIO_InitTypeDef; //类型名为GPIO_InitTypeDef，用此可以定义变量名
```



- 例:初始化PA8

```
//设置PA8为推挽输出，IO口速度为50MHz
void GPIO_Configuration(void)
{ GPIO_InitTypeDef GPIO_InitStructure; //用类型名定义（结构体）变量名
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8; //PA
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度为50MHz
  GPIO_Init(GPIOA, &GPIO_InitStructure); //根据上述参数初始化PA8
```

# 初始化IO口函数 GPIO\_Init

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8;    //PA
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //IO口速度50MHz
GPIO_Init(GPIOA, &GPIO_InitStructure);      //根据上述参数初始化PA8
```

A, B, C, D, E, F, G

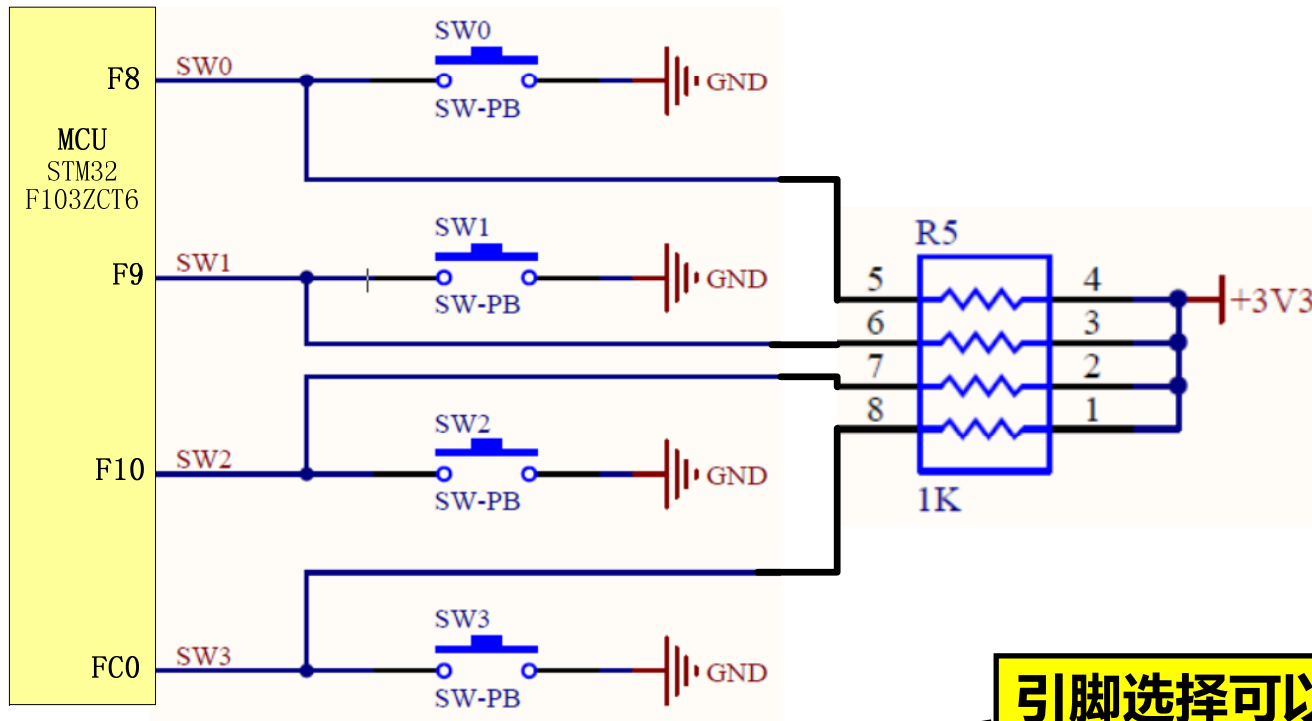
结构体，3个成员变量

GPIO_Mode 值	描述
GPIO_Mode_AIN	模拟输入
GPIO_Mode_IN_FLOATING	浮空输入
GPIO_Mode_IPD	下拉输入
GPIO_Mode_IPU	上拉输入
GPIO_Mode_Out_OD	开漏输出
GPIO_Mode_Out_PP	推挽输出
GPIO_Mode_AF_OD	复用开漏输出
GPIO_Mode_AF_PP	复用推挽输出

GPIO_Speed	描述
GPIO_Speed_10MHz	最高输出速率 10MHz
GPIO_Speed_2MHz	最高输出速率 2MHz
GPIO_Speed_50MHz	最高输出速率 50MHz

GPIO_Pin	描述
GPIO_Pin_None	无管脚被选中
GPIO_Pin_0	选中管脚 0
GPIO_Pin_1	选中管脚 1
GPIO_Pin_2	选中管脚 2
GPIO_Pin_3	选中管脚 3
GPIO_Pin_4	选中管脚 4
GPIO_Pin_5	选中管脚 5
GPIO_Pin_6	选中管脚 6
GPIO_Pin_7	选中管脚 7
GPIO_Pin_8	选中管脚 8
GPIO_Pin_9	选中管脚 9
GPIO_Pin_10	选中管脚 10
GPIO_Pin_11	选中管脚 11
GPIO_Pin_12	选中管脚 12
GPIO_Pin_13	选中管脚 13
GPIO_Pin_14	选中管脚 14
GPIO_Pin_15	选中管脚 15
GPIO_Pin_All	选中全部管脚

## 例：初始化IO口PF8，PF9，PF10



引脚选择可以或操作

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_10; // SW0 ~ SW2
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPD; // 设置成上拉输入
GPIO_Init(GPIOF, &GPIO_InitStructure);
```

# 端口引脚的写操作 GPIO\_WriteBit

- 可用库函数GPIO\_WriteBit对端引脚进行写操作

**GPIO\_WriteBit( GPIO<sub>x</sub>, GPIO\_Pin, BitVal)**

GPIO<sub>x</sub>  
x可以是A, B, C, D, E, F, G

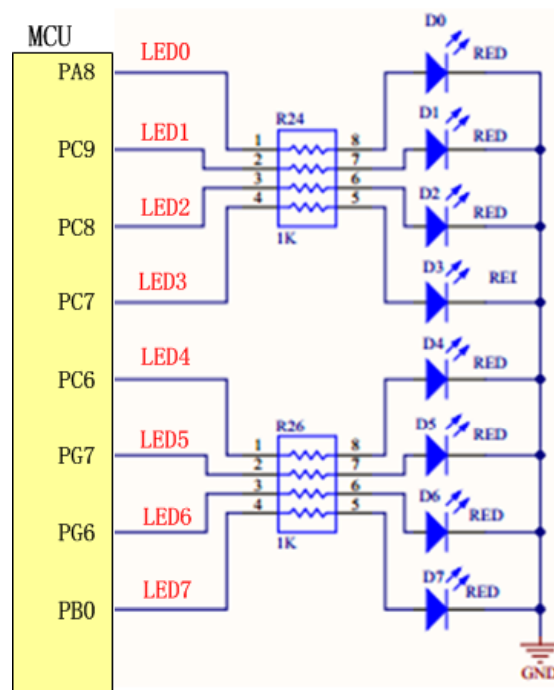
GPIO\_Pin 待设置或者清除的  
端口位：0, 1, 2...15

BitVal枚举类型变量：  
Bit\_SET (设置端口位)  
Bit\_RESET(清除端口位)

//例：点亮LED0

```
GPIO_WriteBit(GPIOA, GPIO_Pin_8, Bit_SET )
```

直接调用函数不便于编程，可读性差，因此，  
往往使用宏定义提高程序的可读性。



# 端口引脚的读操作

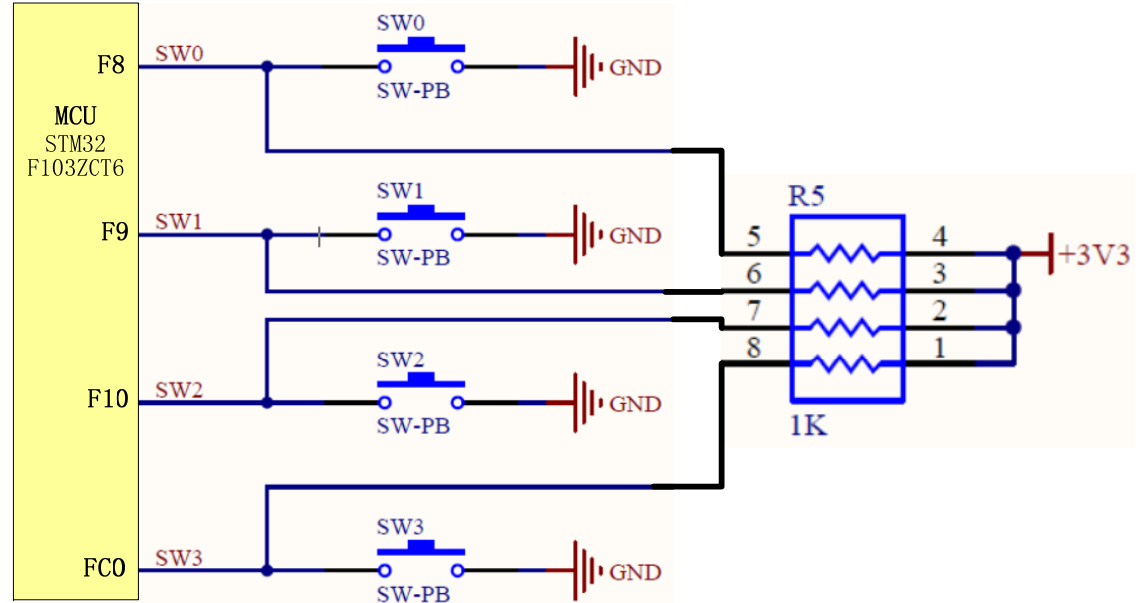
- 可用库函数GPIO\_ReadInputDataBit读端引脚的输入

**GPIO\_ReadInputDataBit(GPIOF,GPIO\_Pin\_8)//读取按键0**

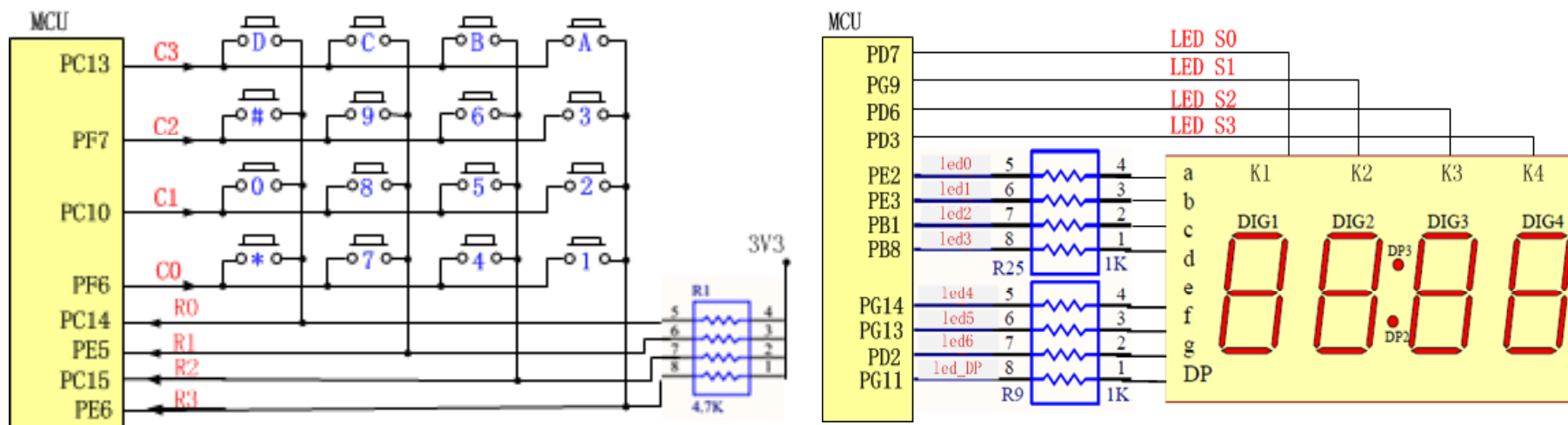
GPIO<sub>x</sub>  
x可以是A, B, C, D, E, F, G

GPIO\_Pin 待读取的  
引脚：0, 1, 2...15

往往使用宏定义  
提高程序的可读性。



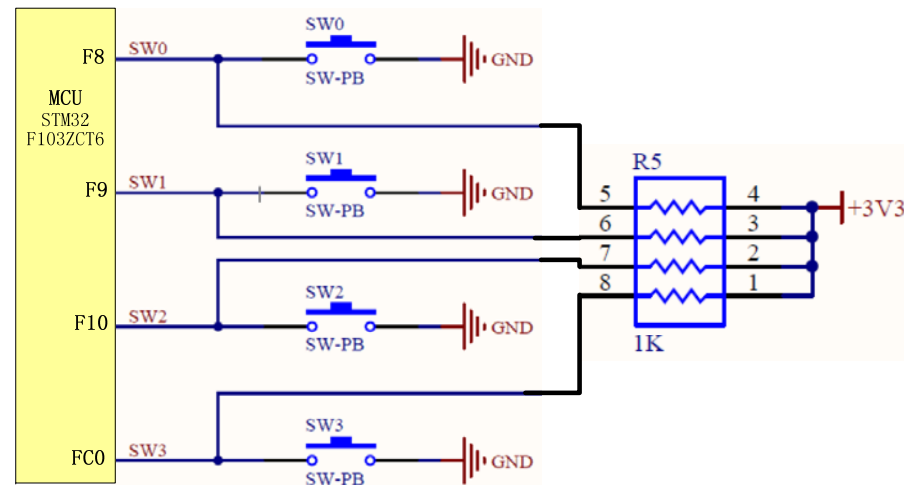
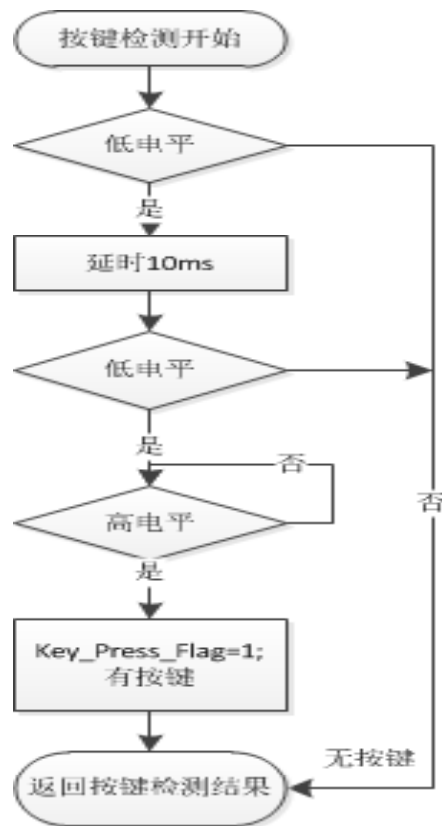
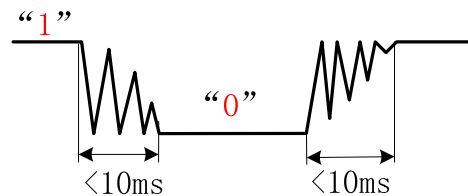
# 例：GPIO\_阵列按键扫描\_数码管显示



- ◆ **程序功能：**扫描矩阵键盘，用数码管显示按键的键值
- ◆ **4\*4矩阵键盘模块：**接口电路设置为行输出，列输入。
- ◆ **键盘扫描方法：**
  - 逐行扫描，分别把每一行置为低电平，其余各行置为高电平；
  - 检查各列线电平的变化，如果某列线电平为低电平，则可确定此行此列交叉点处的按键被按下。
  - 按键位置信息：  $\text{key}[i][j]=1$ （此行此列有键按下）



# 按键检测与消抖动--延时去抖动



```

#define KEY0 GPIO_ReadInputDataBit(GPIOF,GPIO_Pin_8)
.....
char Key_Scan(void)
{
    char Key_pressed_flag=0xff;
    if(KEY0==0)
    {
        Delay1ms(10); //延时10ms
        if(KEY0==0) //检测到按键?
        {
            while(KEY0==0); //等待按键弹起
            Key_pressed_flag=1; //检测到按键
        }
    }
    .....
}
    
```

# 温度数据采集

---

**了解温度测量的方法，掌握STM32F103单片机内部A/D转换的工作原理及数据采集的编程方法。熟悉STM32F103单片机内部A/D转换器的配置方法。**

## ❖ 温度测量工作原理

**STM32F103的ADC中有一个片内温度传感器，温度传感器产生一个与器件内部温度成正比的电压，该电压作为一个单端输入提供给ADC（模/数转换器）的多路选择器。还可以通过另一个ADC温度传感器连接，当选择温度传感器作为ADC的输入并且启动ADC一次转换后，可以通过简单的数学运算将ADC的输出结果转换成相应的温度值。**

## 五. 注意事项

- 严禁自行拆机，以免发生危险
- 严禁频繁进行开、关机的操作
- 严禁在上电状态下进行硬件的插拔
- 严禁用手直接接触芯片，以避免该芯片被静电击穿
- 不要把导线、镊子等工具及书籍纸张、杂物等放在实验箱上。
- 爱护实验设备，实验开发仪还包括众多附件和外设，每次使用完毕后请保持完整，期末实验完毕后由教师验收。



---

谢谢！